

The listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (original) A method of forcibly terminating a thread in a computer language execution environment comprising:
  - a first thread receiving a terminate thread command, the first thread having an associated termination flag, a value of the termination flag being immutable once set and having one or more monitors;
  - setting the termination flag for the first thread;
  - propagating an exception in the execution environment thereby indicating termination of the first thread;
  - ignoring at least one exception handler and a finally clause of the first thread; and
  - exiting one or more monitors associated with the first thread.
2. (original) A method as recited in claim 1 further comprising initiating a termination procedure to cleanly terminate the first thread.
3. (original) A method as recited in claim 1 wherein receiving a terminate thread command further comprises a second thread issuing the termination thread command.
4. (original) A method as recited in claim 1 further comprising determining whether the first thread is in a blocking operation.
5. (original) A method as recited in claim 1 further comprising determining whether a first computer code is part of a user-defined program to be terminated.
6. (original) A method as recited in claim 1 further comprising:
  - associating the one or more monitors locked in the first thread with an execution frame in which the one or more monitors are locked; and
  - exiting the associated one or more monitors when leaving the execution frame.

7. (original) A method as recited in claim 1 further comprising interrupting a monitor lock operation if the monitor lock operation is initiated by a user defined program, the user defined program to be terminated when thread termination is requested.

8. (original) A method as recited in claim 1 wherein the computer language execution environment is the Java programming language execution environment.

9. (original) A method as recited in claim 1 wherein the first thread has an associated termination flag indicator containing an immutable value and an execution field containing a mutable value.

10. (original) A method as recited in claim 1 wherein the terminate thread command is a modified thread.stop command in the Java programming language execution environment.

11. (original) A method as recited in claim 1 further comprising checking a priority of the first thread and, if desired, raising the priority of the thread to avoid delay in terminating the first thread.

12. (original) A method as recited in claim 1 wherein the first thread is cleanly and forcibly terminated in an interpreter loop.

13. (original) A method as recited in claim 12 wherein the interpreter loop is a component of the Java Virtual Machine.

14. (original) A method as recited in claim 1 further comprising determining whether the first thread entered a monitor and whether the first thread has successfully exited the monitor thereby determining whether the first thread has terminated cleanly.

15. (original) An apparatus for forcibly terminating a thread in a computer language execution environment, the apparatus comprising:

means for a first thread to receive a terminate thread command, the first thread having an associated termination flag, a value of the termination flag being immutable once set and having one or more monitors;

means for setting the termination flag for the first thread;

means for propagating an exception in the execution environment thereby indicating termination of the first thread;  
means for ignoring at least one exception handler and a finally clause of the first thread;  
and  
means for exiting one or more monitors associated with the first thread.

16. (original) An apparatus as recited in claim 15 further comprising:  
means for associating the one or more monitors locked in the first thread with an execution frame in which the one or more monitors are locked; and  
means for exiting the associated one or more monitors when leaving the execution frame.
17. (original) An apparatus as recited in claim 15 further comprising means for interrupting a monitor lock operation if the monitor lock operation is initiated by a user defined program, the user defined program to be terminated when thread termination is requested.
18. (original) An apparatus as recited in claim 15 further comprising means for determining whether the first thread entered a monitor and whether the first thread has successfully exited the monitor thereby determining whether the first thread has terminated cleanly.
19. (original) A computer-readable medium containing programmed instructions arranged to forcibly terminate a thread in a computer language execution environment, the computer-readable medium including programmed instructions for:  
a first thread receiving a terminate thread command, the first thread having an associated termination flag, a value of the termination flag being immutable once set and having one or more monitors;  
setting the termination flag for the first thread;  
propagating an exception in the execution environment thereby indicating termination of the first thread;  
ignoring at least one exception handler and a finally clause of the first thread; and  
exiting one or more monitors associated with the first thread.
20. (original) A system for forcibly terminating a thread in a computer language execution environment, the system comprising:

a processor; and  
a computer-readable medium storing a program for execution by the processor, the  
program comprising  
computer code for, at a first thread, receiving a terminate thread command, the  
first thread having an associated termination flag, a value of the termination flag being  
immutable once set and having one or more monitors;  
computer code for setting the termination flag for the first thread;  
computer code for propagating an exception in the execution environment thereby  
indicating termination of the first thread;  
computer code for ignoring at least one exception handler and a finally clause of  
the first thread; and  
computer code for exiting one or more monitors associated with the first thread.